

ATARI 810 DISK
PERIPHERAL DEVICE DESCRIPTION

Preliminary release (9-DEC-80)

Prepared by: Harry B. Stewart
NEOTERIC
15816 San Benito Way
Los Gatos, CA 95030
(408) 395-6478

ATARI 810 DISK
PERIPHERAL DEVICE DESCRIPTION
(preliminary version)

1. Functional description of 810

- 1.1 General description
- 1.2 Overall system performance
 - 1.2.1 Capacity of media
 - 1.2.2 Organization and format of data on media
 - 1.2.3 Maximum and average throughput rates
 - 1.2.4 Operating modes and options
 - 1.2.5 Expandability
- 1.3 Serial bus interface specifications
 - 1.3.1 GET SECTOR
 - 1.3.2 PUT SECTOR
 - 1.3.3 PUT SECTOR WITH VERIFY
 - 1.3.4 STATUS REQUEST
 - 1.3.5 FORMAT DISK
- 1.4 Operating modes and transient behaviors
 - 1.4.1 Power-up
 - 1.4.2 Error handling
 - 1.4.3 Soft RESET
- 1.5 Diagnostic requirements
- 1.6 Configurations/options supported

2. Hardware description

- 2.1 General description
- 2.2 Functional block diagram
- 2.3 Controller memory and I/O address assignments
- 2.4 I/O interface specification
 - 2.4.1 Parallel I/O ports
 - 2.4.1.1 FD1771 Floppy controller ports
 - 2.4.1.2 6532 PIA ports
 - 2.4.2 Serial I/O ports
 - 2.4.3 Interrupts
- 2.5 Electromechanical components
 - 2.5.1 Head stepper motor
 - 2.5.2 Drive motor

3. Firmware description

- 3.1 General description
- 3.2 Command descriptions
- 3.3 Special considerations
- 3.4 Data base description
- 3.5 Module hierarchy
- 3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol
Appendix B -- FD1771 Floppy Disk Controller Chip Description
Appendix C -- 6532 PIA Chip Description
Appendix D -- 6507 Microcomputer Description
Appendix E -- Controller Schematics

1. Functional description of 810

1.1 General description

1.2 Overall system performance

1.2.1 Capacity of media

The Model 810 is designed to accommodate a single single-sided 5 1/4 inch mini-floppy diskette with soft sectoring. Each diskette is formatted to contain 40 tracks of 18 sectors each, with each sector containing 128 data bytes plus overhead; the data capacity of a diskette is thus 92,160 bytes.

1.2.2 Organization and format of data on media

As stated in the preceding paragraph, the disk is formatted to contain 40 tracks, each track containing 18 sectors plus an index field. The outermost track is called track 0 and the innermost track is called track 39. The tracks are separated by xx. The sectors on each track are numbered from 1 to 18 and are physically ordered with an interleave factor of 7, as shown below:

18,7,14,3,10,17,6,13,2,9,16,5,12,1,8,15,4,11

Each sector is comprised of a pre-sector gap, an ID field, a data field and a post-sector gap. The formats for each of these items are provided in the paragraphs that follow.

Index field

The index field is comprised of:

- 256 bytes of zeroes.
- 1 byte of Index Address Mark.
- 6 bytes of zeroes.

The Index Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~xxx~~ is a single byte in length.

Address Mark

Pre-sector gap

The pre-sector gap is comprised of:

- 6 bytes of zeroes.

ID field

The sector ID field is comprised of:

- 1 byte of ID Address Mark.
- 1 byte of track number (\$00-37).
- 1 byte of zeroes.
- 1 byte of sector number (\$01-12).
- 1 byte of zeroes (indicates the sector size = 128).
- 2 bytes of CRC ($G(x) = x^{*16} + x^{*12} + x^{*5} + 1$).

The ID Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark

The CRC is generated by the FD1771 and includes all information starting with the ID Address Mark and up to the CRC characters.

Data field

The sector data field is comprised of:

- 17 bytes of zeroes.
- 1 byte of Data Address Mark.
- 128 bytes of data.
- 2 bytes of CRC ($G(x) = x^{16} + x^{12} + x^5 + 1$).

The Index Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark

The CRC is generated by the FD1771 and includes all information starting with the Index Address Mark and up to the CRC characters.

Post-sector gap

The post-sector gap is comprised of:

- 9 bytes of all zeroes.
- 3 bytes of all ones.

The recording technique used to physically store bits on the media is Frequency Modulation (FM) with a clock pulse being generated for each data bit (except in the case of the special Address Marks); the combined clock pulse and data pulse is referred to as a "cell". The pulses are recorded at a rate of 250,000 pulses per second (1 MHz clock divided by 4) yielding a cell read/write data rate of 125,000 bits per second. Since the data is recorded at fixed frequencies, and since the linear track size decreases as the head is moved toward the center of the disk, the recording density increases on the inner tracks.

1.2.3 Maximum and average throughput rates

Throughput is determined from disk latency + disk transfer time + serial bus transfer time (+ verify for some operations).

1.2.4 Operating modes and options

The Model 810 is to be operational as soon as it is powered on; the controller passively monitors the serial I/O bus looking for a command from the PCS directed to the specific unit set by the switches at the back of the 810. When a period of xx seconds has elapsed since the last command to a given unit, the

controller firmware will step the head to track 39 and turn off the drive motor. When a new command is received for a unit which has the drive motor turned off, the drive motor will be turned on again.

Unit number selection

There are two switches at the back of the unit which provide a logical unit (1-4) selection mechanism; this feature allows up to four 810s to be resident on the serial bus and to be uniquely identified. The logical unit selection is completely independent of the physical unit ordering on the serial bus.

Diskette write protection

There is a sensor in each unit which senses the opacity of a section of the diskette cover. When the section is transparent the disk is not to be written to or formatted; when the section is opaque the disk may be written to or formatted. A disk may always be read.

1.2.5 Expandability

There are no provisions for expansion provided.

1.3 Serial bus interface specifications

This section describes the 810 device specific portions of the serial bus interface; see Appendix A for the general serial bus protocol.

The media is organized by track and sector as explained in section 1.2.2, but the serial bus commands utilize a sector address called the logical sector number, where the sector numbers range from 1 to 720. The sector numbers map to track/sector addresses using the formulae:

$$\begin{aligned}\text{track} &= (\text{logical sector number} - 1) / 18. \\ \text{sector} &= ((\text{logical sector number} - 1) \text{ MOD } 18) + 1.\end{aligned}$$

There are five commands supported by the disk controller:

```
GET SECTOR
PUT SECTOR
PUT SECTOR WITH VERIFY
STATUS REQUEST
FORMAT DISK
```

1.3.1 GET SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.

Command byte = \$52.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then attempts to read the indicated sector and, if successful, returns a COMPLETE byte followed by a data frame consisting of the 128 sector data bytes followed by the frame checksum.

1.3.2 PUT SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$50.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector and finally a COMPLETE byte is sent.

1.3.3 PUT SECTOR WITH VERIFY

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$57.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector, the sector is then verified, and finally a COMPLETE byte is sent.

1.3.4 GET STATUS

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$53.
Auxilliary 1 = N/A.
Auxilliary 2 = N/A.
Command frame checksum.

The controller formats a four byte status frame and sends it to the serial bus preceded by a COMPLETE byte and followed by a checksum of the frame.

The status frame format is shown below:

```

      7              0
+--+--+--+--+--+--+
| command stat. |
+--+--+--+--+--+--+
| hardware stat.|
+--+--+--+--+--+--+
|   timeout   |
+--+--+--+--+--+--+
|   (unused)   |
+--+--+--+--+--+--+

```

The command status contains the following status bits:

Bit-0 = 1 indicates an invalid command frame was received.
 Bit-1 = 1 indicates an invalid data frame was received.
 Bit-2 = 1 indicates that an operation was unsuccessful.
 Bit-3 = 1 indicates that the disk is write protected.
 Bit-4 = 1 indicates active/standby (motor on/off).

The hardware status byte contains the status register of the FD1771 Floppy Disk Controller chip used in the disk controller. See Appendix B for information relating to the meaning of each bit in the byte.

The timeout byte contains a controller provided maximum timeout value (in seconds) to be used by the handler. = xx.

1.3.5 FORMAT DISK

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
 Command byte = \$21.
 Auxilliary 1 = N/A.
 Auxilliary 2 = N/A.
 Command frame checksum.

The controller acknowledges the command frame and then attempts to format the entire disk and to verify it. All bad sector numbers, up to a maximum of 63 numbers, are returned in a standard length data frame; two bytes of all ones (\$FFFF) follow the last bad sector number returned. and put in the supplied buffer, followed by two bytes of all ones (\$FFFF).

1.4 Operating modes and transient behaviors

1.4.1 Power-up

In response to a power-up of the Model 810, the controller

firmware will do the following:

Turn on the drive motor.
Position the head to track 0.
Issue a RESTORE command to the FD1771.
Wait for a command or a drive motor timeout.

1.4.2 Error handling

The following error conditions will be handled as indicated.

Serial bus command frame checksum error -- No response.

Serial bus command frame sector number out of range -- Send NAK.

Serial bus data frame error -- Send NAK.

Write command to protected diskette -- (xx.)

Read error while reading a sector -- (xx.)

Write error while writing a sector -- (xx.)

Inability to find indicated sector (timeout) -- (xx.)

1.4.3 Soft RESET

There is no hardware facility for generating a soft RESET; the unit may be RESET by cycling the Model 810 power ON/OFF switch.

1.5 Diagnostic requirements

There are no requirements for built-in diagnostics for the Model 810.

1.6 Configurations/options supported

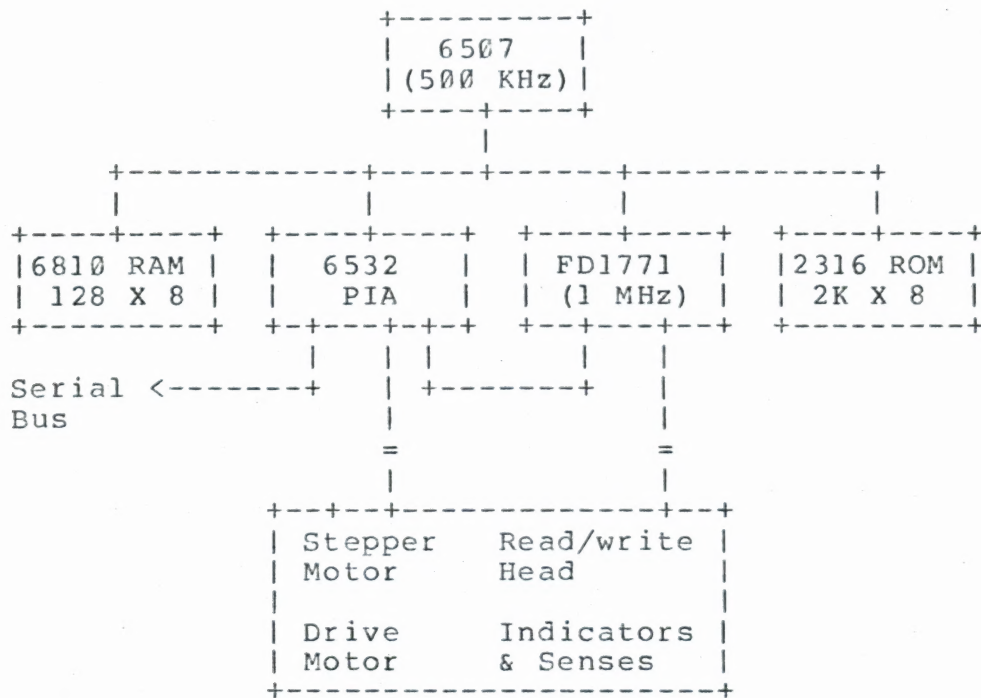
There is a single configuration/option parameter for the Model 810 and that is the logical unit number select. There are a pair of switches at the back of the unit which are used to indicate which of the four possible logical units (1-4) this unit is to become. These switches are to be checked as part of the processing of each command frame.

2. Hardware description

2.1 General description

The Model 810 is comprised of a (xx) mini-floppy diskette drive and an Atari designed and built controller board mounted together in a single package.

A block diagram of the controller is shown below:



2.2 Functional block diagram

2.3 Controller memory and I/O address assignments

ROM address space = 0800-0FFF (2316).
RAM address space = 0080-00FF (6801)
 0180-01FF (6532).
I/O address space = 0000-0003 (FD1771)
 0380-039F (6532).

2.4 I/O interface specification

2.4.1 Parallel I/O ports

All of the I/O for the controller processor is performed via two sets of mapped memory I/O ports; one set is assigned to a FD1771 Floppy Disk Controller chip and the other set is assigned to a 6532 PIA chip. The ports are described further in the sections that follow.

2.4.1.1 FD1771 Floppy Disk Controller chip

The FD1771 chip data access lines expect and provide data in bit complemented form (1 for 0 and 0 for 1); since there is no hardware inversion of the data bus between the 6507 and the FD1771, the inversion must be performed by the controller firmware. The descriptions in this section, as well as those to be found in Appendix B, describe the data as seen inside the FD1771, which is the complement of the data as seen by the 6507.

The port address assignments are shown below:

Note that many functions of chip not used.

Address	Direction	Function
0000	READ	Status register.
0000	WRITE	Command register.
0001	RD/WRT	Track register.
0002	RD/WRT	Sector register.
0003	READ	Read data register.
0003	WRITE	Write data register.

The description of the FD1771 chip is to be found in Appendix B.

2.4.1.2 6532 Peripheral Interface Adaptor (PIA)

The port address assignments are shown below:

Address	Direction	Function
0380	RD/WRT	Port A.
0381	WRITE	Port A data direction register.
0382	RD/WRT	Port B.
0383	WRITE	Port B data direction register.
0384	(WRITE)	Negative edge detect, disable int.
0385	(WRITE)	Positive edge detect, disable int.
0385	READ	Read and clear interrupt flag.
0386	(WRITE)	Negative edge detect, enable int.
0387	(WRITE)	Positive edge detect, enable int.
039X	(WRITE)	Interval timer setup.

The bit assignments to the 6532 are shown below, by port:

Port	Bit-#	Direction	Function
0380	7	READ	1 = data request from FD1771.
	6	READ	1 = FD1771 IRQ.
	5	N/A	unused.
	4	READ	1 = diskette write protected.
	3	N/A	unused.
	2	READ	Drive select switch #2 (Note 1).
	1	WRITE	1 = drive motor on/drive select.
	0	READ	Drive select switch #1 (Note 1).
0381	7-0	WRITE	Port 0380 data direction select: 1 = PIA output pin, 0 = PIA input pin.

Note 1 -- The drive select switch values are mapped to logical unit numbers as shown in the table below:

Switch #2 Switch #1 Logical unit

1	1	1
1	0	2
0	0	3
0	1	4

Port	Bit-#	Direction	Function
0382	7	READ	Serial bus data input (inverted).
	6	READ	Serial bus NOT COMMAND- (1=command).
	5	WRITE	Stepper motor phase 4 (Note 2).
	4	WRITE	Stepper motor phase 3 (Note 2).
	3	WRITE	Stepper motor phase 2 (Note 2).
	2	WRITE	Stepper motor phase 1 (Note 2).
	1	READ	Serial bus +5V/READY (0=READY).
	0	WRITE	Serial bus data output (data true). (Note 3).
0383	7-0	WRITE	Port 0382 data direction select: 1 = PIA output pin, 0 = PIA input pin.

Note 2 -- See section 2.5.1 for a description of the stepper motor characteristics.

Note 3 -- The serial bus data output line is to be switched from a PIA output line to an input line whenever the controller is not outputting data to the serial bus.

The ports below all relate to internal functions of the 6532 PIA; the ports designated '(WRITE)' do not operate upon the write data, the command is entirely encoded in the address (as explained in Appendix C).

Port	Bit-#	Direction	Function
0384	(WRITE)		Negative edge detect, disable int.
0385	(WRITE)		Positive edge detect, disable int.
0385	READ		Read and clear interrupt flag.
0386	(WRITE)		Negative edge detect, enable int.
0387	(WRITE)		Positive edge detect, enable int.
039X	(WRITE)		Interval timer setup.

The description of the 6532 chip is to be found in Appendix C.

2.4.2 Serial I/O ports

There are no general purpose serial I/O ports provided in the Model 810.

2.4.3 Interruts

There is no interrupt capability in the 6507 processor chip.

2.5 Electromechanical components

2.5.1 Head stepper motor

The continuous stepper motor sequence for head inward movement is shown below; the sequence in reverse order provides head outward movement:

Phase4	Phase3	Phase2	Phase1
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1
1	1	0	0
etc.			

Other combinations such as 0 0 0 0, etc. do what?

The minimum delay between steps for reliable operation is 5.25 milliseconds.

The reference track is track 0, which may be reached by stepping outward a calculated number of tracks from a known track or by stepping outward some arbitrarily large number that guarantees that track zero is reached. No harm is done by attempting to drive the stepper beyond track 0.

The inter-track spacing is xx which is equivalent to 1 step.

2.5.2 Diskette drive motor

The operational speed for the drive motor is xx RPM.

The time required for the motor to get to operational speed from a dead stop is 200 milliseconds.

3. Firmware description

3.1 General description

3.2 Command descriptions

3.3 Special considerations

3.4 Data base description

3.5 Module hierarchy

3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol

This appendix describes the electrical characteristics of the Atari 400/800 serial bus, the use of the bus to send bytes of data, the organization of the bytes as "frames" (records), and the overall command sequences which utilize frames and response bytes to provide computer/peripheral communication. All signal direction terminology assumes the viewpoint of the PCS computer (e.g. in/out, receive/send, etc.).

Hardware/electrical characteristics

The Atari 400/800 computer communicates with peripheral devices over a 19,200 baud asynchronous serial port. The serial port consists of a serial DATA OUT (transmission) line, a serial DATA IN (receiver) line and other miscellaneous control lines.

Data is transmitted and received as 8 bits of serial data (LSB sent first) preceded by a logic zero start bit and succeeded by a logic one stop bit. The serial DATA OUT is transmitted as positive logic (+4v = one/true/high, 0v = zero/false/low). The serial DATA OUT line always assumes its new state when the serial CLOCK OUT line goes high; CLOCK OUT then goes low in the center of the DATA OUT bit time.

An end view of the serial bus connector at the computer or peripheral is shown below (the cable connectors would of course be a mirror image):

	2	4	6	8	10	12
	o	o	o	o	o	o
o	o	o	o	o	o	o
1	3	5	7	9	11	13

where: 1 = computer CLOCK IN.
 2 = computer CLOCK OUT.
 3 = computer DATA IN.
 4 = GND.
 5 = computer DATA OUT.
 6 = GND.
 7 = COMMAND-.
 8 = MOTOR CONTROL.
 9 = PROCEED-.
 10 = +5v/READY.
 11 = computer AUDIO IN.
 12 = +12v.
 13 = INTERRUPT-.

1 CLOCK IN is not used by the present O.S. and peripherals. This line can be used in future synchronous communications schemes.

- 2 CLOCK OUT is the serial bus clock. CLOCK OUT goes high at the start of each DATA OUT bit and returns to low in the middle of each bit.
- 3 DATA IN is the serial bus data line to the computer.
- 4 Pin 4 GND is the signal/shield ground line.
- 5 DATA OUT is the serial bus data line from the computer.
- 6 Pin 6 GND is the signal/shield ground line.
- 7 COMMAND- is normally high and goes low when a command frame is being sent from the computer.
- 8 MOTOR CONTROL is the cassette motor control line (high=on, low= off).
- 9 PROCEED- is not used by the present O.S. and peripherals (this line is pulled high).
- 10 +5v/READY indicates that the computer is turned on and ready. This line may also be used as a +5 volt supply of ~~unknown current rating~~ for Atari peripherals only.
- 11 AUDIO IN accepts an audio signal from the cassette.
- 12 Pin 12 is a +12 volt supply of ~~unknown current rating~~ for Atari peripherals only.
- 13 INTERRUPT- is not used by the present O.S. and peripherals (this line is pulled high).

There are no pin reassignments made in the serial bus cable, so pin 3, the computer's DATA IN line, is the peripheral's data output line; and similarly for pin 5.

Serial bus electrical specifications

Peripheral input:

V_{LH} = 2.0v min.

V_{LL} = 0.4v max.

I_{LH} = 20ua. max. @ V_{LH} = 2.0v

I_{LL} = 5ua. max. @ V_{LL} = .4v

Peripheral output (open collector bipolar):

V_{OL} = 0.4v max. @ 1.6 ma.

V_{OH} = 4.5v min. with external 100Kohm pull-up.

V_{cc}/READY input:

V_{LH} = 2.0v min. @ I_{LH} = 1ma. max.

V_{LL} = 0.4v max.

Input goes to logic zero when open.

Bus commands

The bus protocol specifies that all commands must originate from the computer, and that peripherals will present data on the bus only when commanded to. Every bus operation will go to completion before another bus operation is initiated (no overlap). An error detected at any point in the command sequence will abort the entire sequence.

A bus operation consists of the following elements:

- Command frame from the computer.

- Acknowledgement (ACK) from the peripheral.

- Optional data frame to or from the computer.

- Operation complete (COMPLETE) from the peripheral.

COMMAND FRAME

The serial bus protocol provides for three types of commands: 1) data send, 2) data receive and 3) immediate (no data -- command only). There is a common element in all three types, a command frame consisting of five bytes of information sent from the computer while the COMMAND- line is held low. The format of the command frame is shown below:

```
+-----+
| device I.D. |
+-----+
|  command  |
+-----+
| auxilliary #1 |
+-----+
| auxilliary #2 |
+-----+
|  checksum  |
+-----+
```

The device I.D. specifies which of the serial bus devices is being addressed.

The command byte contains a device dependent command.

The auxilliary bytes contain more device dependent information.

The checksum byte contains the arithmetic sum of the first four bytes (with the carry added back after every addition).

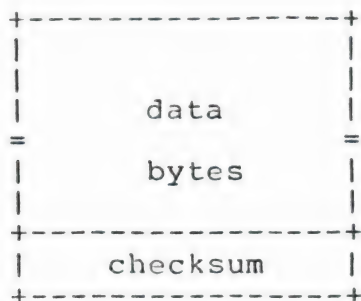
COMMAND FRAME ACKNOWLEDGE

The peripheral being addressed would normally respond to a command frame by sending an ACK byte (\$41) to the computer; if there is a problem with the command frame, the peripheral should not respond.

DATA FRAME

Following the command frame (and ACK) may be an optional data

frame which is formatted as shown below:



This data frame may originate at the computer or at the device controller, depending upon the command. Current device controllers expect fixed length data frames as does the computer, where the data frame length is a fixed function of the device type and command.

The checksum value in the data frame is the arithmetic sum of all of the frame data preceding the checksum, with the carry from each addition being added back (the same algorithm as for the command frame).

In the case of the computer sending a data frame to a peripheral, the peripheral is expected to send an ACK if the data frame is acceptable, and a NAK (\$4E) or nothing if the data frame is unacceptable.

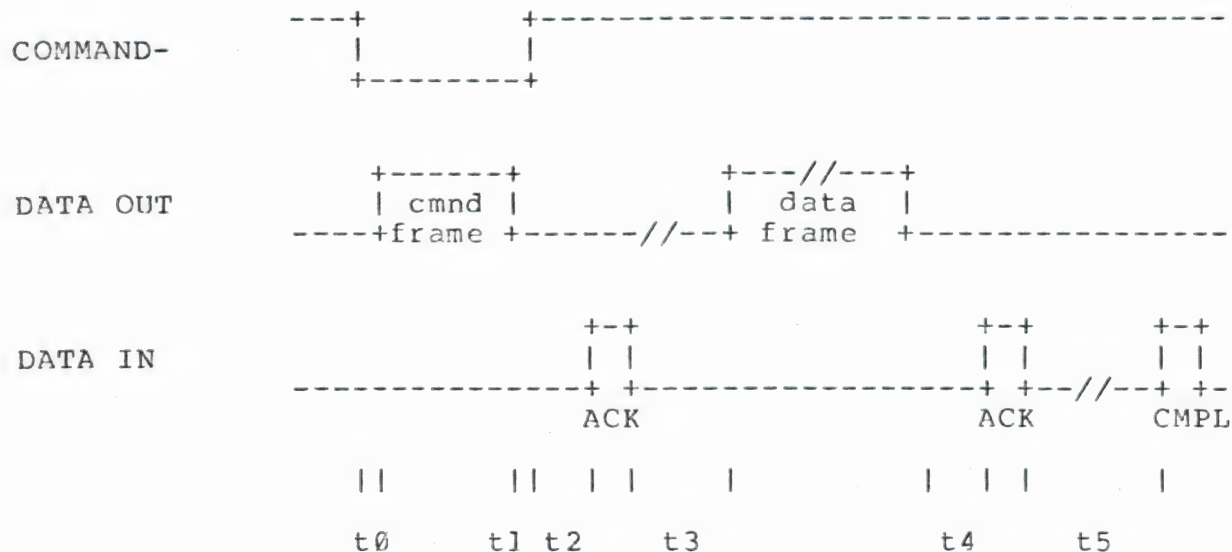
OPERATION COMPLETE

A peripheral is also expected to send an operation COMPLETE byte (\$43) at the time the commanded operation is complete. The location of this byte in the command sequence for each command type is shown in the timing diagrams that follow. If the operation cannot go to normal, error-free completion, the peripheral should respond with an ERROR byte (\$45) instead of COMPLETE.

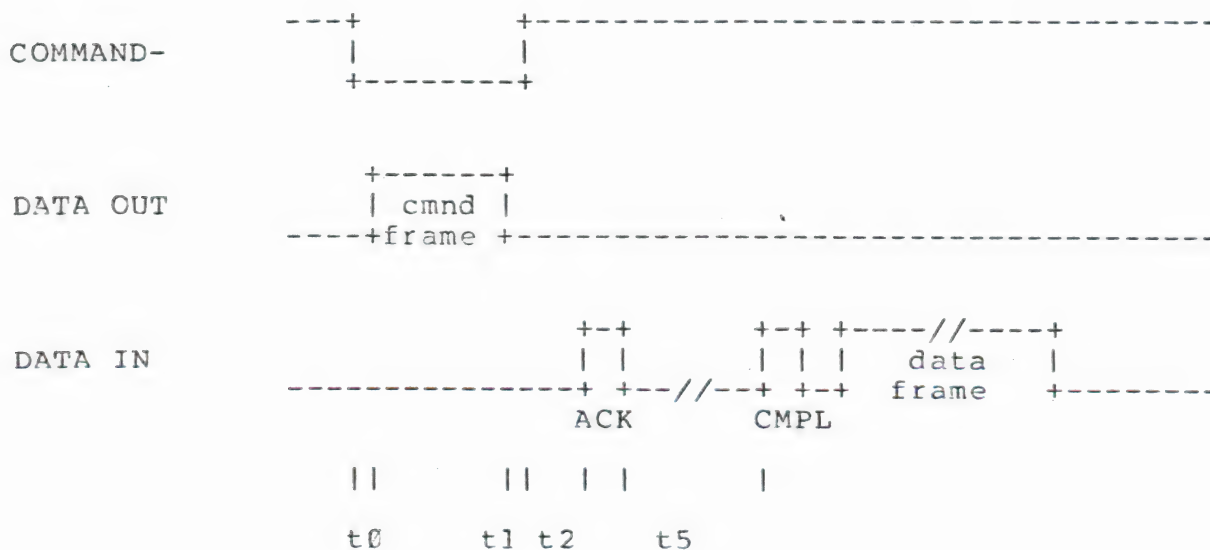
Bus timing

This section provides timing diagrams for the three types of command sequences: data send, data receive and immediate.

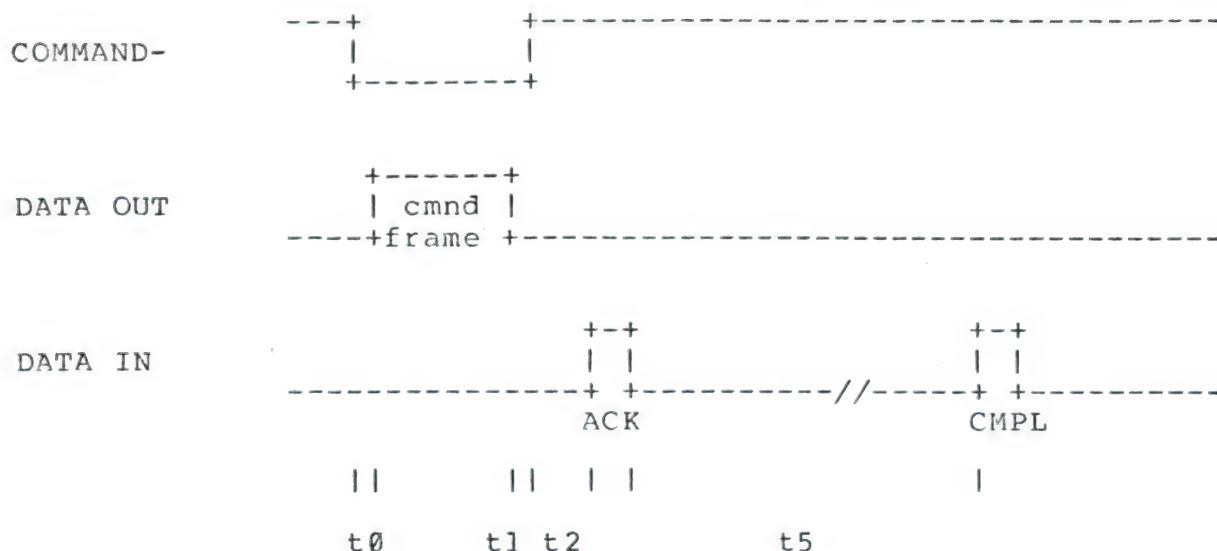
COMPUTER DATA SEND (PERIPHERAL DATA RECEIVE) sequence:



COMPUTER DATA RECEIVE (PERIPHERAL DATA SEND) sequence:



IMMEDIATE sequence:



t0 is the delay between the lowering of COMMAND- and the transmission of the first byte of the command frame. The computer generates this delay.

computer t0 (min) = 750 usec.
computer t0 (max) = 1600 usec.

peripheral t0 (min) = ??
peripheral t0 (max) = ??

t1 is the delay between the transmission of the last bit of the command frame and the raising of the COMMAND- line. This delay is generated by the computer.

computer t1 (min) = 650 usec.
computer t1 (max) = 950 usec.

peripheral t1 (min) = ??
peripheral t1 (max) = ??

t2 is the delay between the raising of COMMAND- and the transmission of the ACK byte by the peripheral. The peripheral generates this delay.

computer t2 (min) = 0 usec.
computer t2 (max) = 16 msec.

peripheral t2 (min) = ??
peripheral t2 (max) = ??

t3 is the delay between the receipt of the last bit of the ACK byte and the transmission of the first bit of the data frame by the computer. The computer generates this delay.

computer t3 (min) = 1000 usec.
computer t3 (max) = 1800 usec.

peripheral t3 (min) = ??
peripheral t3 (max) = ??

t4 is the delay between the transmission of the last bit of the data frame and the receipt of the first bit of the ACK byte by the computer. The peripheral generates this delay.

computer t4 (min) = 850 usec.
computer t4 (max) = 16 msec.

peripheral t4 (min) = ??
peripheral t4 (max) = ??

t5 is the delay between the receipt of the last bit of the ACK byte and the first bit of the COMPLETE byte by the computer. The peripheral generates this delay.

computer t5 (min) = 250 usec.
computer t5 (max) = 255 sec. (handler dependent)

peripheral t5 (min) = ??
peripheral t5 (max) = N/A

Serial bus special character values

ACK = \$41.
NAK = \$4E.
COMPLETE = \$43.
ERR = \$45.

Serial bus device I.D. to logical unit mapping (Model 810)

Unit 1 = \$31.
Unit 2 = \$32.
Unit 3 = \$33.
Unit 4 = \$34.

controller flow charts from O.S. manual

Appendix B -- FD1771 Floppy Disk Controller Chip Description

Appendix C -- 6532 PIA Chip Description

Appendix D -- 6507 Microcomputer Description

Appendix E -- Model 810 Controller Schematics

(011 SPEC)

Dave Stanger

ATARI 810 DISK
PERIPHERAL DEVICE DESCRIPTION

Preliminary release

(9-DEC-80)

Please review and
return.

B-J.

Info on stepper & drive etc.

Prepared by: Harry B. Stewart
NEOTERIC
15816 San Benito Way
Los Gatos, CA 95030
(408) 395-6478

ATARI 810 DISK
PERIPHERAL DEVICE DESCRIPTION
(preliminary version) ✓

1. Functional description of 810

- 1.1 General description
- 1.2 Overall system performance
 - 1.2.1 Capacity of media
 - 1.2.2 Organization and format of data on media
 - 1.2.3 Maximum and average throughput rates
 - 1.2.4 Operating modes and options
 - 1.2.5 Expandability
- 1.3 Serial bus interface specifications
 - 1.3.1 GET SECTOR
 - 1.3.2 PUT SECTOR
 - 1.3.3 PUT SECTOR WITH VERIFY
 - 1.3.4 STATUS REQUEST
 - 1.3.5 FORMAT DISK
- 1.4 Operating modes and transient behaviors
 - 1.4.1 Power-up
 - 1.4.2 Error handling
 - 1.4.3 Soft RESET
- 1.5 Diagnostic requirements
- 1.6 Configurations/options supported

2. Hardware description

- ✓ 2.1 General description
- 2.2 ~~Functional~~ Block diagram
- 2.3 Controller memory and I/O address assignments
- 2.4 I/O interface specification
 - 2.4.1 Parallel I/O ports
 - 2.4.1.1 FD1771 Floppy controller ports
 - 2.4.1.2 6532 PIA ports
 - 2.4.2 Serial I/O ports
 - 2.4.3 Interrupts
- 2.5 Electromechanical components
 - 2.5.1 Head stepper motor
 - 2.5.2 Drive motor

3. Firmware description

- 3.1 General description
- 3.2 Command descriptions
- 3.3 Special considerations
- 3.4 Data base description
- 3.5 Module hierarchy
- 3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol
Appendix B -- FD1771 Floppy Disk Controller Chip Description
Appendix C -- 6532 PIA Chip Description
Appendix D -- 6507 Microcomputer Description
Appendix E -- Controller Schematics

Appendix F - MPI Diskette Drive Description ✓

1. Functional description of 810

1.1 General description

1.2 Overall system performance

1.2.1 Capacity of media

The Model 810 is designed to accommodate a single single-sided 5 1/4 inch mini-floppy diskette with soft sectoring. Each diskette is formatted to contain 40 tracks of 18 sectors each, with each sector containing 128 data bytes plus overhead; the data capacity of a diskette is thus 92,160 bytes.

1.2.2 Organization and format of data on media

As stated in the preceding paragraph, the disk is formatted to contain 40 tracks, each track containing 18 sectors plus an index field. The outermost track is called track 0 and the innermost track is called track 39. The tracks are separated by xx. The sectors on each track are numbered from 1 to 18 and are physically ordered with an interleave factor of 7, as shown below:

18,7,14,3,10,17,6,13,2,9,16,5,12,1,8,15,4,11

Each sector is comprised of a pre-sector gap, an ID field, a data field and a post-sector gap. The formats for each of these items are provided in the paragraphs that follow.

Index field

The index field is comprised of:

- 256 bytes of zeroes.
- 1 byte of Index Address Mark.
- 6 bytes of zeroes.

The Index Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~mark~~ is a single byte in length.

Address Mark

Pre-sector gap

The pre-sector gap is comprised of:

- 6 bytes of zeroes.

ID field

The sector ID field is comprised of:

- 1 byte of ID Address Mark.
- 1 byte of track number (00-39). *0-39 DECIMAL*
- 1 byte of zeroes. *(SIDE NUMBER)*
- 1 byte of sector number (01-12).
- 1 byte of zeroes (indicates the sector size = 128).
- 2 bytes of CRC ($G(x) = x^{16} + x^{12} + x^5 + 1$).

The ID Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark

The ^{ID field} CRC is generated by the FD1771 and includes all information starting with the ID Address Mark and up to the ² CRC characters. BYTES

Data field

The sector data field is comprised of:

- 17 bytes of zeroes. ^{PRE-DATA FIELD GAP}
- 1 byte of Data Address Mark.
- 128 bytes of data.
- 2 bytes of CRC ($G(x) = x^{16} + x^{12} + x^5 + 1$). ^{DATA}

The Index Address Mark is generated by the FD1771 ^{IN} is response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark

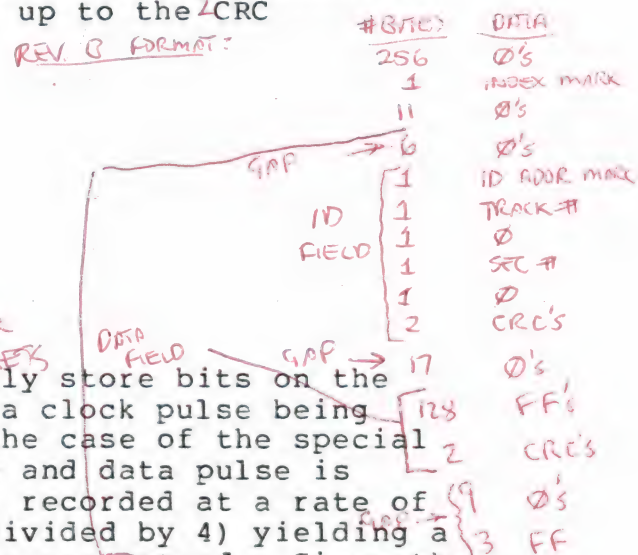
The ^{DATA} CRC is generated by the FD1771 and includes all information starting with the Index Address Mark and up to the ² CRC characters. BYTES

Post-sector gap

The post-sector gap is comprised of:

- 9 bytes of all zeroes.
- 3 bytes of all ones.

The recording technique used to physically store bits on the media is Frequency Modulation (FM) with a clock pulse being generated for each data bit (except in the case of the special Address Marks); the combined clock pulse and data pulse is referred to as a "cell". The pulses are recorded at a rate of 250,000 pulses per second (1 MHz clock divided by 4) yielding a cell read/write data rate of 125,000 bits per second. Since the data is recorded at fixed frequencies, and since the linear track size decreases as the head is moved toward the center of the disk, the recording density increases on the inner tracks.



1.2.3 Maximum and average throughput rates

Throughput is determined from disk latency + disk transfer time + serial bus transfer time (+ verify for some operations).

1.2.4 Operating modes and options

The Model 810 is to be operational as soon as it is powered on; the controller passively monitors the serial I/O bus looking for a command from the PCS directed to the specific unit set by the switches at the back of the 810. When a period of xx seconds has elapsed since the last command to a given unit, the

controller firmware will step the head to track 39 and turn off the drive motor. When a new command is received for a unit which has the drive motor turned off, the drive motor will be turned on again.

Unit number selection

There are two switches at the back of the unit which provide a logical unit (1-4) selection mechanism; this feature allows up to four 810s to be resident on the serial bus and to be uniquely identified. The logical unit selection is completely independent of the physical unit ordering on the serial bus.

Diskette write protection

There is a sensor in each unit which senses the opacity of a section of the diskette cover. When the section is transparent the disk is not to be written to or formatted; when the section is opaque the disk may be written to or formatted. A disk may always be read.

✓ TRANSPARENT
✓ OPAQUE
WHEN THE WRITE PROTECT TAB COVERS A NOTCH IN THE FLOPPY DISKETTE COVER, THE CONTROLLER FIRMWARE WILL NOT PERMIT DISK WRITE OR FORMAT OPERATIONS TO OCCUR ON THAT MEDIA

1.2.5 Expandability

There are no provisions for expansion provided.

1.3 Serial bus interface specifications

This section describes the 810 device specific portions of the serial bus interface; see Appendix A for the general serial bus protocol.

The media is organized by track and sector as explained in section 1.2.2, but the serial bus commands utilize a sector address called the logical sector number, where the sector numbers range from 1 to 720. The sector numbers map to track/sector addresses using the formulae:

INTEGER PART OF 2 ✓

$$\begin{aligned}\text{track} &= (\text{logical sector number} - 1) / 18. \\ \text{sector} &= ((\text{logical sector number} - 1) \text{ MOD } 18) + 1.\end{aligned}$$

There are five commands supported by the disk controller:

GET SECTOR
PUT SECTOR
PUT SECTOR WITH VERIFY
STATUS REQUEST
FORMAT DISK

1.3.1 GET SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.

Command byte = \$52.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then attempts to read the indicated sector and, if successful, returns a COMPLETE byte followed by a data frame consisting of the 128 sector data bytes followed by the frame checksum.

oops! THIS GOES IN "ERROR HANDLING"
IF THE COMMAND FRAME REFERENCES THE GIVEN CONTROLLER AND A ~~ERROR~~ ^{KEY} IS PP AND A CMD FRAME CHECKSUM ERROR IS DETECTED, A NAK IS SENT. IF AN OUT OF RANGE LOGICAL SECTOR # IS RECEIVED, A NAK IS SENT. IF THE CONTROLLER CANNOT COMPLETE THE READ AN ERROR (\$45) BYTE IS SENT.

1.3.2 PUT SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$50.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector and finally a COMPLETE byte is sent.

1.3.3 PUT SECTOR WITH VERIFY

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$57.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector, the sector is then verified, and finally a COMPLETE byte is sent.

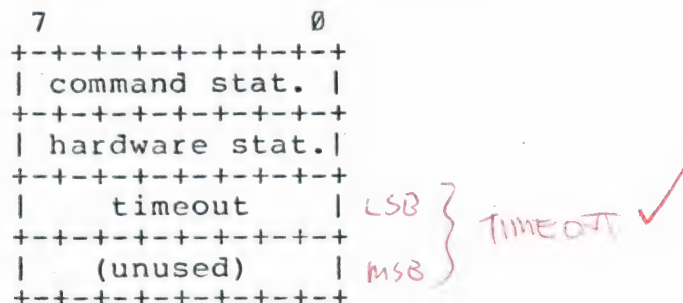
1.3.4 GET STATUS

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$53.
Auxilliary 1 = N/A.
Auxilliary 2 = N/A.
Command frame checksum.

The controller formats a four byte status frame and sends it to the serial bus preceded by a COMPLETE byte and followed by a checksum of the frame.

The status frame format is shown below:



The command status contains the following status bits:

- Bit-0 = 1 indicates an invalid command frame was received.
- Bit-1 = 1 indicates an invalid data frame was received.
- Bit-2 = 1 indicates that an operation was unsuccessful.
- Bit-3 = 1 indicates that the disk is write protected.
- Bit-4 = 1 indicates active/standby (motor on/off).

The hardware status byte contains the status register of the FD1771 Floppy Disk Controller chip used in the disk controller. See Appendix B for information relating to the meaning of each bit in the byte.

INVERTED ✓
 FOR THE LAST 1771 COMMAND ISSUED. ✓

The timeout byte contains a controller provided maximum timeout value (in seconds) to be used by the handler.

? ✓

1.3.5 FORMAT DISK

The controller receives a command frame containing the following information:

- Device I.D. = \$31-34.
- Command byte = \$21.
- Auxilliary 1 = N/A.
- Auxilliary 2 = N/A.
- Command frame checksum.

The controller acknowledges the command frame and then attempts to format the entire disk and to verify it. All bad sector numbers, up to a maximum of 63 numbers, are returned in a standard length data frame; two bytes of all ones (\$FFFF) follow the last bad sector number returned. and put in the supplied buffer, followed by two bytes of all ones (\$FFFF).

LOGICAL ✓
 ✓

STANDARD 128 BITE ✓

1.4 Operating modes and transient behaviors

1.4.1 Power-up

In response to a power-up of the Model 810, the controller

firmware will do the following:

Turn on the drive motor.
Position the head to track 0.
~~Issue a RESTORE command to the FDI771.~~
Wait for a command or a drive motor timeout.

- TURN ON DRIVE MOTOR
- NAME THE HEAD TO TRACK #0
BY STEPPING OFF 39X
- WAIT FOR A COMMAND
OR AFTER ABOUT 7 SECONDS
STEP IN TO TRACK #39 AND
TURN OFF MOTOR

1.4.2 Error handling

The following error conditions will be handled as indicated.

Serial bus command frame checksum error -- No response. *if UNIT # INCORRECT, ELSE NAK SENT*

Serial bus command frame sector number out of range -- Send NAK

Serial bus data frame *CHECKSUM* error -- Send NAK.

Write command to protected diskette -- **(xx.)** *ACK CMD FRAME, THEN (F45) ERROR BYTE SENT*

Read error while reading a sector -- **(xx.)** *ERROR BYTE SENT AFTER 4 TRIALS*

Write error while writing a sector -- **(xx.)** *ERROR BYTE SENT AFTER 4 TRIALS*

Inability to find indicated sector (timeout) -- **(xx.)** *HEAD IS POSITIONED ON SUPPOSED TRACK, AND IF SECTOR IS NOT FOUND IN 500 MS, OPERATION IS RETRIED. FOUR TRIALS ARE PERFORMED, AND AFTER THE 3RD TRIAL, A HOME TO TRACK #0, THEN A RESEEK TO SUPPOSED TRACK IS PERFORMED. ERROR BYTE SENT AFTER 4 TRIALS*

1.4.3 Soft RESET

There is no hardware facility for generating a soft RESET; the unit may be RESET by cycling the Model 810 power ON/OFF switch.

1.5 Diagnostic requirements

There are no requirements for built-in diagnostics for the Model 810.

1.6 Configurations/options supported

There is a single configuration/option parameter for the Model 810 and that is the logical unit number select. There are a pair of switches at the back of the unit which are used to indicate which of the four possible logical units (1-4) this unit is to become. These switches are to be checked as part of the processing of each command frame.

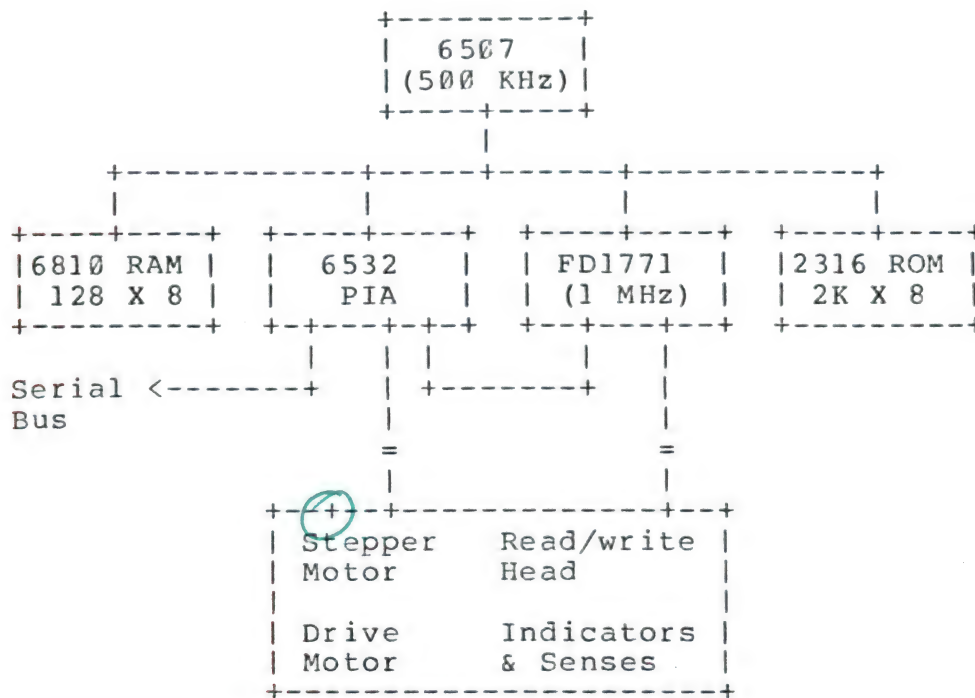
- ✓ ~~FDI771~~
- ✓ ERROR WHILE FORMATTING A TRACK (TAKES LONGER THAN 500 MS) -- ERROR BYTE SENT, THEN 128 BYTE BAD SECTOR BUFFER
- ✓ ANY SECTORS BAD ON FORMAT VERIFY -- ERROR BYTE SENT, THEN 128 BYTE BAD SECTOR BUFFER (2 TRIALS FOR SECTOR READ BACK)
- ✓ READ AFTER WRITE ERROR -- (READ FAILURE AFTER 2 TRIALS OR COMPARE ERROR) ERROR BYTE SENT

2. Hardware description

2.1 General description

The Model 810 is comprised of a (xx) mini-floppy diskette drive and an Atari designed and built controller board mounted together in a single package.

A block diagram of the controller is shown below:



2.2 Functional Block diagram

2.3 Controller memory and I/O address assignments

ROM address space = 0800-0FFF (2316).
RAM address space = 0080-00FF (6801)
 0180-01FF (6532).
I/O address space = 0000-0003 (FD1771)
 0380-039F (6532).

~~I/O IS PERFORMED VIA 2 SETS OF MAPPED MEMORY PORTS.~~

2.4 I/O interface specification

2.4.1 Parallel I/O ports

All of the I/O for the controller processor is performed via two sets of mapped memory I/O ports; one set is assigned to a FD1771 Floppy Disk Controller chip and the other set is assigned to a 6532 PIA chip. The ports are described further in the sections that follow.

2.4.1.1 FD1771 Floppy Disk Controller chip

COMMUNICATIONS WITH 1771 ARE ACCOMPLISHED VIA MEMORY MAPPED ADDRESSES 0-3. A 6532 PIA CHIP ALLOWS SOFTWARE UART COMMUNICATIONS VIA SERIAL BUS AND SOFTWARE CONTROL OF STEPPER MOTOR, WRITE PROTECT STOPS, DRIVE SELECT, ETC.

The FD1771 chip data access lines expect and provide data in bit complemented form (1 for 0 and 0 for 1); since there is no hardware inversion of the data bus between the 6507 and the FD1771, the inversion must be performed by the controller firmware. The descriptions in this section, as well as those to be found in Appendix B, describe the data as seen inside the FD1771, which is the complement of the data as seen by the 6507.

The *memory mapped* port address assignments are shown below:

Address	Direction	Function
0000	READ	Status register.
0000	WRITE	Command register.
0001	RD/WRT	Track register.
0002	RD/WRT	Sector register.
0003	READ	Read data register.
0003	WRITE	Write data register.

The description of the FD1771 chip is to be found in Appendix B.

2.4.1.2 6532 Peripheral Interface Adaptor (PIA)

The port address assignments are shown below:

Address	Direction	Function
0380	RD/WRT	Port A.
0381	WRITE	Port A data direction register.
0382	RD/WRT	Port B.
0383	WRITE	Port B data direction register.
0384	(WRITE)	Negative edge detect, disable int.
0385	(WRITE)	Positive edge detect, disable int.
0385	READ	Read and clear interrupt flag.
0386	(WRITE)	Negative edge detect, enable int.
0387	(WRITE)	Positive edge detect, enable int.
039X	(WRITE)	Interval timer setup.

The bit assignments to the 6532 are shown below, by port:

Port	Bit-#	Direction	Function
0380	7	READ	1 = data request from FD1771.
	6	READ	1 = FD1771 IRQ.
	5	N/A	unused.
	4	READ	1 = diskette write protected.
	3	N/A	unused.
	2	READ	Drive select switch #2 (Note 1).
	1	WRITE	1 = drive motor on/drive select.
	0	READ	Drive select switch #1 (Note 1).
0381	7-0	WRITE	Port 0380 data direction select: 1 = PIA output pin, 0 = PIA input pin.

Note 1 -- The drive select switch values are mapped to logical unit numbers as shown in the table below:

Switch #2 Switch #1 Logical unit

1	1	1
1	0	2
0	0	3
0	1	4

Port	Bit-#	Direction	Function
0382	7	READ	Serial bus data input (inverted).
	6	READ	Serial bus NOT COMMAND- (1=command).
	5	WRITE	Stepper motor phase 4 (Note 2).
	4	WRITE	Stepper motor phase 3 (Note 2).
	3	WRITE	Stepper motor phase 2 (Note 2).
	2	WRITE	Stepper motor phase 1 (Note 2).
	1	READ	Serial bus +5V/READY (0=READY).
	0	WRITE	Serial bus data output (data true). (Note 3).
0383	7-0	WRITE	Port 0382 data direction select: 1 = PIA output pin, 0 = PIA input pin.

Note 2 -- See section 2.5.1 for a description of the stepper motor characteristics.

Note 3 -- The serial bus data output line is to be switched from a PIA output line to an input line whenever the controller is not outputting data to the serial bus.

The ports below all relate to internal functions of the 6532 PIA; the ports designated '(WRITE)' do not operate upon the write data, the command is entirely encoded in the address (as explained in Appendix C).

Port	Bit-#	Direction	Function
0384	(WRITE)		Negative edge detect, disable int.
0385	(WRITE)		Positive edge detect, disable int.
0385	READ		Read and clear interrupt flag.
0386	(WRITE)		Negative edge detect, enable int.
0387	(WRITE)		Positive edge detect, enable int.
039X	(WRITE)		Interval timer setup.

The description of the 6532 chip is to be found in Appendix C.

2.4.2 Serial I/O ports

There are no general purpose serial I/O ports provided in the Model 810.

2.4.3 Interrupts

There is no interrupt capability in the 6507 processor chip.

2.5 Electromechanical components

2.5.1 Head stepper motor

The continuous stepper motor sequence for head inward movement is shown below; the sequence in reverse order provides head outward movement:

Phase4	Phase3	Phase2	Phase1	TRACK #'s
1	1	0	0	$\phi \text{ mod } 4$
0	1	1	0	$1 \text{ mod } 4$
0	0	1	1	$2 \text{ mod } 4$
1	0	0	1	$3 \text{ mod } 4$
$\bar{1}$	1	0	0	$0 \text{ mod } 4$
etc.				

Other combinations such as 0 0 0 0, etc. do what?

PHASE BITS ARE ZEROED
ON MOTOR SHUTDOWN ✓

The minimum delay between steps for reliable operation is 5.25 milliseconds.

The reference track is track 0, which may be reached by stepping outward a calculated number of tracks from a known track or by stepping outward some arbitrarily large number that guarantees that track zero is reached. No harm is done by attempting to drive the stepper beyond track 0. NOT CLEAR ✓

The inter-track spacing is $\boxed{xx}$ which is equivalent to 1 step.

48 TRACKS/INCH ? ✓

2.5.2 Diskette drive motor

305 ? ✓

The operational speed for the drive motor is $\boxed{xx}$ RPM.

The time required for the motor to get to operational speed from a dead stop is 200 milliseconds.

4 9 39

39 = 1001

→ back 43 steps

3. Firmware description

3.1 General description

3.2 Command descriptions

3.3 Special considerations

3.4 Data base description

3.5 Module hierarchy

3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol

This appendix describes the electrical characteristics of the Atari 400/800 serial bus, the use of the bus to send bytes of data, the organization of the bytes as "frames" (records), and the overall command sequences which utilize frames and response bytes to provide computer/peripheral communication. All signal direction terminology assumes the viewpoint of the PCS computer (e.g. in/out, receive/send, etc.).

Hardware/electrical characteristics

The Atari 400/800 computer communicates with peripheral devices over a 19,200 baud asynchronous serial port. The serial port consists of a serial DATA OUT (transmission) line, a serial DATA IN (receiver) line and other miscellaneous control lines.

Data is transmitted and received as 8 bits of serial data (LSB sent first) preceded by a logic zero start bit and succeeded by a logic one stop bit. The serial DATA OUT is transmitted as positive logic (+4v = one/true/high, 0v = zero/false/low). The serial DATA OUT line always assumes its new state when the serial CLOCK OUT line goes high; CLOCK OUT then goes low in the center of the DATA OUT bit time.

An end view of the serial bus connector at the computer or peripheral is shown below (the cable connectors would of course be a mirror image):

2	4	6	8	10	12	
o	o	o	o	o	o	
o	o	o	o	o	o	
1	3	5	7	9	11	13

where: 1 = computer CLOCK IN.
2 = computer CLOCK OUT.
3 = computer DATA IN.
4 = GND.
5 = computer DATA OUT.
6 = GND.
7 = COMMAND-.
8 = MOTOR CONTROL.
9 = PROCEED-.
10 = +5v/READY.
11 = computer AUDIO IN.
12 = +12v.
13 = INTERRUPT-.

- 1 CLOCK IN is not used by the present O.S. and peripherals. This line can be used in future synchronous communications schemes.

- 2 CLOCK OUT is the serial bus clock. CLOCK OUT goes high at the start of each DATA OUT bit and returns to low in the middle of each bit.
- 3 DATA IN is the serial bus data line to the computer.
- 4 Pin 4 GND is the signal/shield ground line.
- 5 DATA OUT is the serial bus data line from the computer.
- 6 Pin 6 GND is the signal/shield ground line.
- 7 COMMAND- is normally high and goes low when a command frame is being sent from the computer.
- 8 MOTOR CONTROL is the cassette motor control line (high=on, low= off).
- 9 PROCEED- is not used by the present O.S. and peripherals (this line is pulled high).
- 10 +5v/READY indicates that the computer is turned on and ready. This line may also be used as a +5 volt supply of ~~unknown current rating~~ for Atari peripherals only. *50 ma* ✓] ✓
- 11 AUDIO IN accepts an audio signal from the cassette.
- 12 Pin 12 is a +12 volt supply of ~~unknown current rating~~ for Atari peripherals only.] ✓
- 13 INTERRUPT- is not used by the present O.S. and peripherals (this line is pulled high).

There are no pin reassignments made in the serial bus cable, so pin 3, the computer's DATA IN line, is the peripheral's data output line; and similarly for pin 5.

Serial bus electrical specifications

Peripheral input:

V_{LH} = 2.0v min.

V_{LL} = 0.4v max.

I_{LH} = 20ua. max. @ V_{LH} = 2.0v

I_{LL} = 5ua. max. @ V_{LL} = .4v

Peripheral output (open collector bipolar):

V_{OL} = 0.4v max. @ 1.6 ma.

V_{OH} = 4.5v min. with external 100Kohm pull-up.

V_{CC}/READY input:

V_{LH} = 2.0v min. @ I_{LH} = 1ma. max.

V_{LL} = 0.4v max.

Input goes to logic zero when open.

Bus commands

The bus protocol specifies that all commands must originate from the computer, and that peripherals will present data on the bus only when commanded to. Every bus operation will go to completion before another bus operation is initiated (no overlap). An error detected at any point in the command sequence will abort the entire sequence.

A bus operation consists of the following elements:

Command frame from the computer.

Acknowledgement (ACK) from the peripheral.

Optional data frame to or from the computer.

Operation complete (COMPLETE) from the peripheral.

COMMAND FRAME

The serial bus protocol provides for three types of commands: 1) data send, 2) data receive and 3) immediate (no data -- command only). There is a common element in all three types, a command frame consisting of five bytes of information sent from the computer while the COMMAND- line is held low. The format of the command frame is shown below:

+-----+
device I.D.
+-----+
command
+-----+
auxilliary #1
+-----+
auxilliary #2
+-----+
checksum
+-----+

The device I.D. specifies which of the serial bus devices is being addressed.

The command byte contains a device dependent command.

The auxilliary bytes contain more device dependent information.

The checksum byte contains the arithmetic sum of the first four bytes (with the carry added back after every addition).

COMMAND FRAME ACKNOWLEDGE

The peripheral being addressed would normally respond to a command frame by sending an ACK byte (\$41) to the computer; if there is a problem with the command frame, the peripheral should not respond.

DATA FRAME

Following the command frame (and ACK) may be an optional data

frame which is formatted as shown below:



This data frame may originate at the computer or at the device controller, depending upon the command. Current device controllers expect fixed length data frames as does the computer, where the data frame length is a fixed function of the device type and command.

The checksum value in the data frame is the arithmetic sum of all of the frame data preceding the checksum, with the carry from each addition being added back (the same algorithm as for the command frame).

In the case of the computer sending a data frame to a peripheral, the peripheral is expected to send an ACK if the data frame is acceptable, and a NAK (\$4E) or nothing if the data frame is unacceptable.

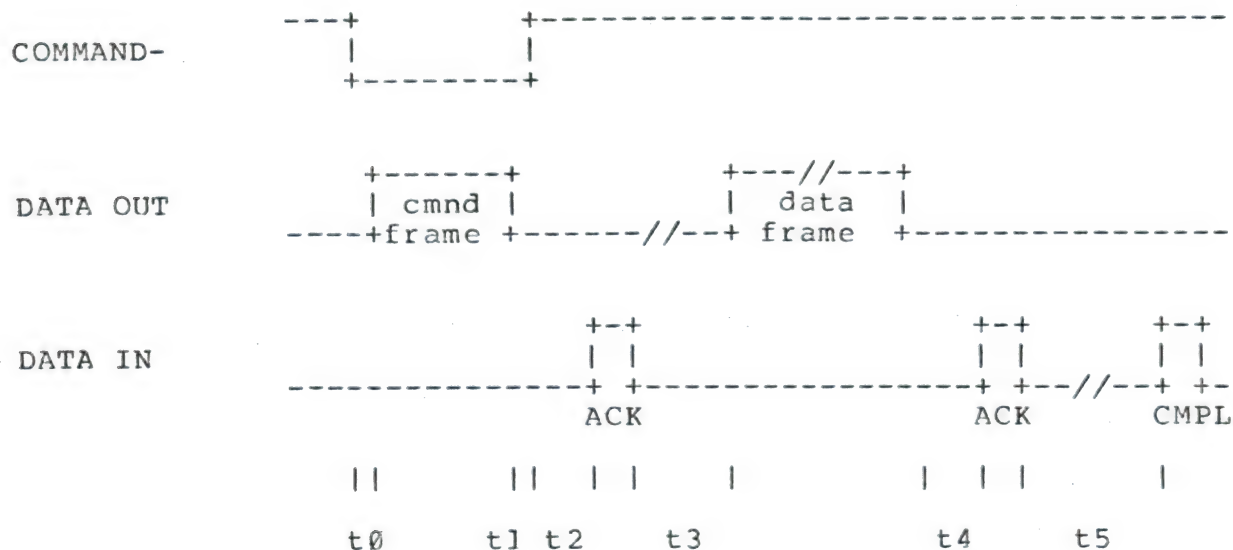
OPERATION COMPLETE

A peripheral is also expected to send an operation COMPLETE byte (\$43) at the time the commanded operation is complete. The location of this byte in the command sequence for each command type is shown in the timing diagrams that follow. If the operation cannot go to normal, error-free completion, the peripheral should respond with an ERROR byte (\$45) instead of COMPLETE.

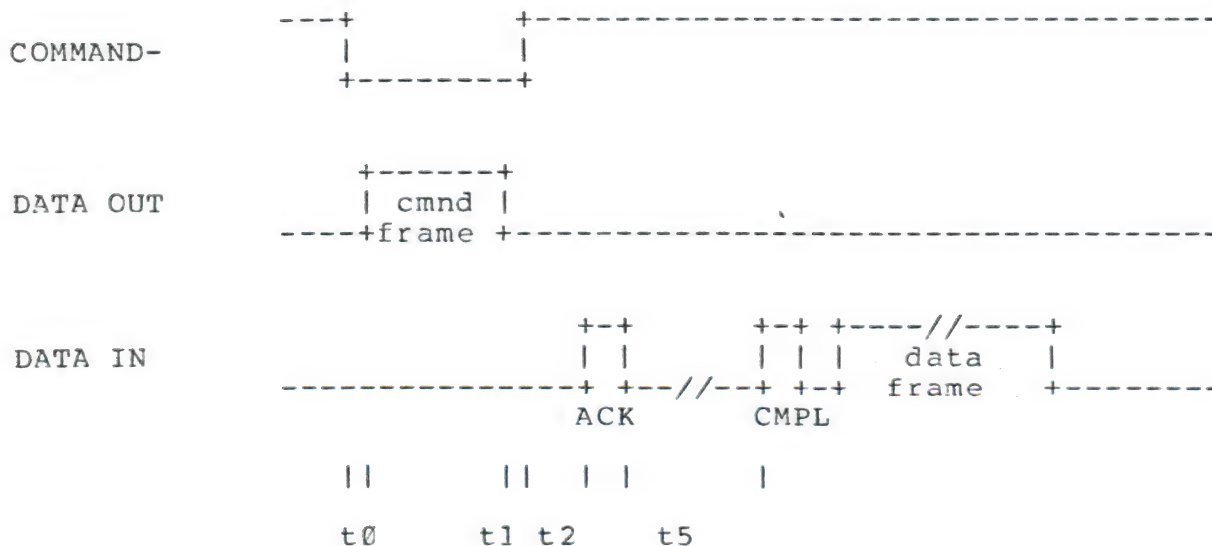
Bus timing

This section provides timing diagrams for the three types of command sequences: data send, data receive and immediate.

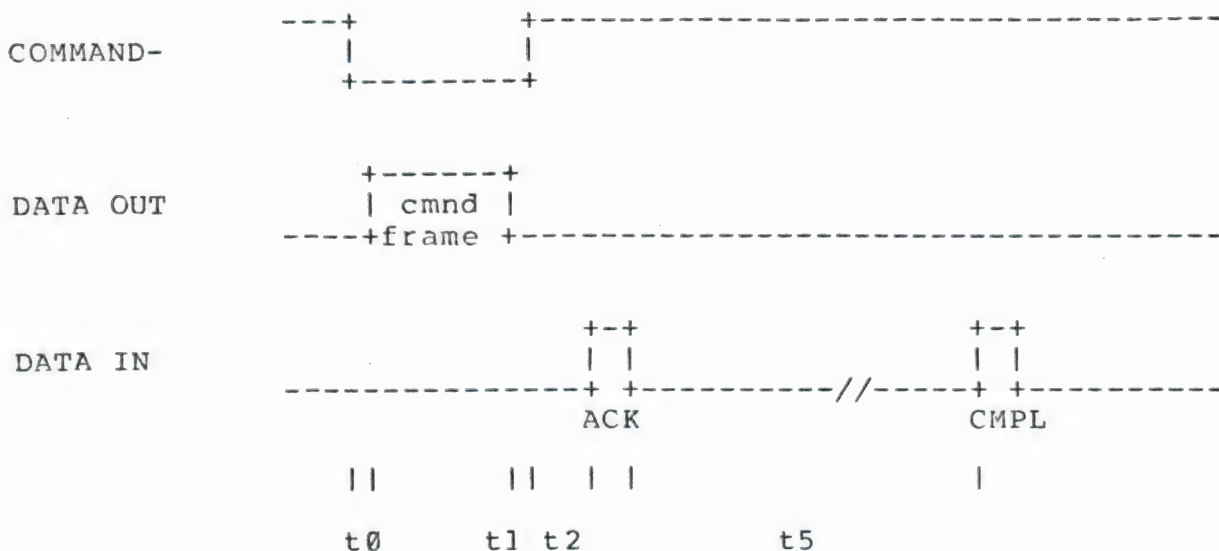
COMPUTER DATA SEND (PERIPHERAL DATA RECEIVE) sequence:



COMPUTER DATA RECEIVE (PERIPHERAL DATA SEND) sequence:



IMMEDIATE sequence:



t0 is the delay between the lowering of COMMAND- and the transmission of the first byte of the command frame. The computer generates this delay.

computer t0 (min) = 750 usec.
computer t0 (max) = 1600 usec.

peripheral t0 (min) = ??
peripheral t0 (max) = ??

t1 is the delay between the transmission of the last bit of the command frame and the raising of the COMMAND- line. This delay is generated by the computer.

computer t1 (min) = 650 usec.
computer t1 (max) = 950 usec.

peripheral t1 (min) = ??
peripheral t1 (max) = ??

t2 is the delay between the raising of COMMAND- and the transmission of the ACK byte by the peripheral. The peripheral generates this delay.

computer t2 (min) = 0 usec.
computer t2 (max) = 16 msec.

peripheral t2 (min) = ??
peripheral t2 (max) = ??

t3 is the delay between the receipt of the last bit of the ACK byte and the transmission of the first bit of the data frame by the computer. The computer generates this delay.

computer t3 (min) = 1000 usec.
computer t3 (max) = 1800 usec.

peripheral t3 (min) = ??
peripheral t3 (max) = ??

t4 is the delay between the transmission of the last bit of the data frame and the receipt of the first bit of the ACK byte by the computer. The peripheral generates this delay.

computer t4 (min) = 850 usec.
computer t4 (max) = 16 msec.

peripheral t4 (min) = ??
peripheral t4 (max) = ??

t5 is the delay between the receipt of the last bit of the ACK byte and the first bit of the COMPLETE byte by the computer. The peripheral generates this delay.

computer t5 (min) = 250 usec.
computer t5 (max) = 255 sec. (handler dependent)

peripheral t5 (min) = ??
peripheral t5 (max) = N/A

Serial bus special character values

ACK = \$41.
NAK = \$4E.
COMPLETE = \$43.
ERR = \$45.

Serial bus device I.D. to logical unit mapping (Model 810)

Unit 1 = \$31.
Unit 2 = \$32.
Unit 3 = \$33.
Unit 4 = \$34.

Appendix B -- FD1771 Floppy Disk Controller Chip Description

Appendix C -- 6532 PIA Chip Description

Appendix D -- 6507 Microcomputer Description

(page)

Appendix F -- MPI Discrete Drive Description



Scott. Input

ATARI 810 DISK PERIPHERAL DEVICE DESCRIPTION

Preliminary release

(9-DEC-80)

found 1
" ~~13~~ 17 18
" ~~39~~ 40 42

Error handling? ?

Talk about diff. controller
software versions &
distinguish each at
appropriate points in
text?

Software: I (incomplete)

Hardware: ~~incomplete~~ (I) incomplete:

add timing diagrams,
signal characteristics on
diskette, tolerances, etc.
Schematics

Prepared by: Harry B. Stewart
NEOTERIC
15816 San Benito Way
Los Gatos, CA 95030
(408) 395-6478

ATARI 810 DISK
PERIPHERAL DEVICE DESCRIPTION
~~(preliminary version)~~

1. Functional description of 810

- 1.1 General description
- 1.2 Overall system performance
 - 1.2.1 Capacity of media
 - 1.2.2 Organization and format of data on media
 - 1.2.3 Maximum and average throughput rates
 - 1.2.4 Operating modes and options
 - 1.2.5 Expandability
- 1.3 Serial bus interface specifications
 - 1.3.1 GET SECTOR
 - 1.3.2 PUT SECTOR
 - 1.3.3 PUT SECTOR WITH VERIFY
 - 1.3.4 STATUS REQUEST
 - 1.3.5 FORMAT DISK
- 1.4 Operating modes and transient behaviors
 - 1.4.1 Power-up
 - 1.4.2 Error handling
 - 1.4.3 Soft RESET
- 1.5 Diagnostic requirements
- 1.6 Configurations/options supported

2. Hardware description

- 2.1 General description
- 2.2 Functional block diagram
- 2.3 Controller memory and I/O address assignments
- 2.4 I/O interface specification
 - 2.4.1 Parallel I/O ports
 - 2.4.1.1 FD1771 Floppy controller ports
 - 2.4.1.2 6532 PIA ports
 - 2.4.2 Serial I/O ports
 - 2.4.3 Interrupts
- 2.5 Electromechanical components
 - 2.5.1 Head stepper motor
 - 2.5.2 Drive motor

3. Firmware description

- 3.1 General description
- 3.2 Command descriptions
- 3.3 Special considerations
- 3.4 Data base description
- 3.5 Module hierarchy
- 3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol
Appendix B -- FD1771 Floppy Disk Controller Chip Description
Appendix C -- 6532 PIA Chip Description
Appendix D -- 6507 Microcomputer Description
Appendix E -- Controller Schematics

1. Functional description of 810

1.1 General description

1.2 Overall system performance

1.2.1 Capacity of media

The Model 810 is designed to accommodate a single single-sided 5 1/4 inch mini-floppy diskette with soft sectoring. Each diskette is formatted to contain 40 tracks of 18 sectors each, with each sector containing 128 data bytes plus overhead; the data capacity of a diskette is thus 92,160 bytes.

1.2.2 Organization and format of data on media

As stated in the preceding paragraph, the disk is formatted to contain 40 tracks, each track containing 18 sectors plus an index field. The outermost track is called track 0 and the innermost track is called track 39. The tracks are separated by xx. The sectors on each track are numbered from 1 to 18 and are physically ordered with an interleave factor of 7, as shown below:

18,7,14,3,10,17,6,13,2,9,16,5,12,1,8,15,4,11

Each sector is comprised of a pre-sector gap, an ID field, a data field and a post-sector gap. The formats for each of these items are provided in the paragraphs that follow.

Index field

The index field is comprised of:

- 256 bytes of zeroes.
- 1 byte of Index Address Mark.
- 6 bytes of zeroes.

The Index Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~is~~ is a single byte in length.

Pre-sector gap

The pre-sector gap is comprised of:

- 6 bytes of zeroes.

ID field

The sector ID field is comprised of:

- 1 byte of ID Address Mark.
- 1 byte of track number (\$00-37).
- 1 byte of zeroes.
- 1 byte of sector number (\$01-12).
- 1 byte of zeroes (indicates the sector size = 128).
- 2 bytes of CRC ($G(x) = x^{16} + x^{12} + x^5 + 1$).

terminology

\$XX ✓
COMMAND-

centers ✓

1/4 inch ✓

Is this some form of IBM "standard" format? If so, say so, & what the exceptions are.

define. ✓

Address Mark ✓

(as in "Space")

zeros = no data or written zeros (i.e. definable bytes)?
Be more specific.

plus or xor?

The ID Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark ✓

The CRC is generated by the FD1771 and includes all information starting with the ID Address Mark and up to the CRC characters.

Data field

The sector data field is comprised of:

- 17 bytes of zeroes.
- 1 byte of Data Address Mark.
- 128 bytes of data.
- 2 bytes of CRC ($G(x) = x^{16} + x^{12} + x^5 + 1$).

The ~~Index~~ Data Address Mark is generated by the FD1771 in response to a command from the controller to do so. The ~~IAM~~ is a single byte in length. Address Mark ✓

The CRC is generated by the FD1771 and includes all information starting with the Index Address Mark and up to the CRC characters.

Post-sector gap

The post-sector gap is comprised of:

- 9 bytes of all zeroes.
- 3 bytes of all ones.

do pre-sector gaps
& post-sector gaps
have anything between?
If so, what? noise?
how about stop? How
much? where?

The recording technique used to physically store bits on the media is Frequency Modulation (FM) with a clock pulse being generated for each data bit (except in the case of the special Address Marks); the combined clock pulse and data pulse is referred to as a "cell". The pulses are recorded at a rate of 250,000 pulses per second (1 MHz clock divided by 4) yielding a cell read/write data rate of 125,000 bits per second. Since the data is recorded at fixed frequencies, and since the linear track size decreases as the head is moved toward the center of the disk, the recording density increases on the inner tracks.

explain FM. Any control bits (ie, stop, start?)

1.2.3 Maximum and average throughput rates

Throughput is determined from disk latency + disk transfer time + serial bus transfer time (+ verify for some operations).

1.2.4 Operating modes and options

The Model 810 is to be operational as soon as it is powered on; the controller passively monitors the serial I/O bus looking for a command from the PCS directed to the specific unit set by the switches at the back of the 810. When a period of xx seconds has elapsed since the last command to a given unit, the

controller firmware will step the head to track 39 and turn off the drive motor. When a new command is received for a unit which has the drive motor turned off, the drive motor will be turned on again. *Even if command will cause no disk I/O?*

Unit number selection

There are two switches at the back of the unit which provide a logical unit (1-4) selection mechanism; this feature allows up to four 810s to be resident on the serial bus and to be uniquely identified. The logical unit selection is completely independent of the physical unit ordering on the serial bus.

Diskette write protection

There is a sensor in each unit which senses the opacity of a section of the diskette cover. When the section is transparent the disk is not to be written to or formatted; when the section is opaque the disk may be written to or formatted. A disk may always be read. *! (BACK-WARDS)*

1.2.5 Expandability *special*

There are no provisions for expansion provided.

(How about retrofit of controller software?)

1.3 Serial bus interface specifications

This section describes the 810 device specific portions of the serial bus interface; see Appendix A for the general serial bus protocol.

The media is organized by track and sector as explained in section 1.2.2, but the serial bus commands utilize a sector address called the logical sector number, where the sector numbers range from 1 to 720. The sector numbers map to track/sector addresses using the formulae:

$$\begin{aligned}\text{track} &= (\text{logical sector number} - 1) / 18. \\ \text{sector} &= ((\text{logical sector number} - 1) \text{ MOD } 18) + 1.\end{aligned}$$

There are five commands supported by the disk controller:

GET SECTOR
PUT SECTOR
PUT SECTOR WITH VERIFY
STATUS REQUEST
FORMAT DISK

1.3.1 GET SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.

*define terminology ✓
(\$ means ...)*

Command byte = \$52.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then attempts to read the indicated sector and, if successful, returns a COMPLETE byte followed by a data frame consisting of the 128 sector data bytes followed by the frame checksum.

1.3.2 PUT SECTOR

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$50.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector and finally a COMPLETE byte is sent.

1.3.3 PUT SECTOR WITH VERIFY

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$57.
Auxilliary 1 = sector number (lsb).
Auxilliary 2 = sector number (msb).
Command frame checksum.

The controller then acknowledges the command frame and waits for a data frame. When the data frame is received, it is first acknowledged and then the frame data is written to the specified sector, the sector is then verified, and finally a COMPLETE byte is sent.

1.3.4 GET STATUS

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$53.
Auxilliary 1 = N/A.
Auxilliary 2 = N/A.
Command frame checksum.

(C1W SPEC)

The controller formats a four byte status frame and sends it to the serial bus preceded by a COMPLETE byte and followed by a checksum of the frame.

The status frame format is shown below:

```

      7                                0
+--+--+--+--+--+--+--+--+--+
| command stat. |
+--+--+--+--+--+--+--+--+--+
| hardware stat. |
+--+--+--+--+--+--+--+--+--+
|      timeout      |
+--+--+--+--+--+--+--+--+--+
|      (unused)      |
+--+--+--+--+--+--+--+--+--+
```

The command status contains the following status bits:

Bit-0 = 1 indicates an invalid command frame was received.
Bit-1 = 1 indicates an invalid data frame was received.
Bit-2 = 1 indicates that an operation was unsuccessful.
Bit-3 = 1 indicates that the disk is write protected.
Bit-4 = 1 indicates active/standby (motor on/off).

The hardware status byte contains the status register of the FD1771 Floppy Disk Controller chip used in the disk controller. See Appendix B for information relating to the meaning of each bit in the byte.

The timeout byte contains a controller provided maximum timeout value (in seconds) to be used by the handler.

1.3.5 FORMAT DISK

The controller receives a command frame containing the following information:

Device I.D. = \$31-34.
Command byte = \$21.
Auxilliary 1 = N/A.
Auxilliary 2 = N/A.
Command frame checksum.

The controller acknowledges the command frame and then attempts to format the entire disk and to verify it. All bad sector numbers, up to a maximum of 63 numbers, are returned in a standard length data frame; two bytes of all ones (\$FFFF) follow the last bad sector number returned. and put in the supplied buffer, followed by two bytes of all ones (\$FFFF).

1.4 Operating modes and transient behaviors

1.4.1 Power-up

In response to a power-up of the Model 810, the controller

firmware will do the following:

Turn on the drive motor.
Position the head to track 0.
Issue a RESTORE command to the FD1771.
Wait for a command or a drive motor timeout.

1.4.2 Error handling

The following error conditions will be handled as indicated.

Serial bus command frame checksum error -- No response.

Serial bus command frame sector number out of range -- Send NAK.

Serial bus data frame error -- Send NAK.

Write command to protected diskette -- (xx.)

Read error while reading a sector -- (xx.)

Write error while writing a sector -- (xx.)

Inability to find indicated sector (timeout) -- (xx.)

1.4.3 Soft RESET

There is no hardware facility for generating a soft RESET; the unit may be RESET by cycling the Model 810 power ON/OFF switch.

1.5 Diagnostic requirements

There are no requirements for built-in diagnostics for the Model 810.

1.6 Configurations/options supported

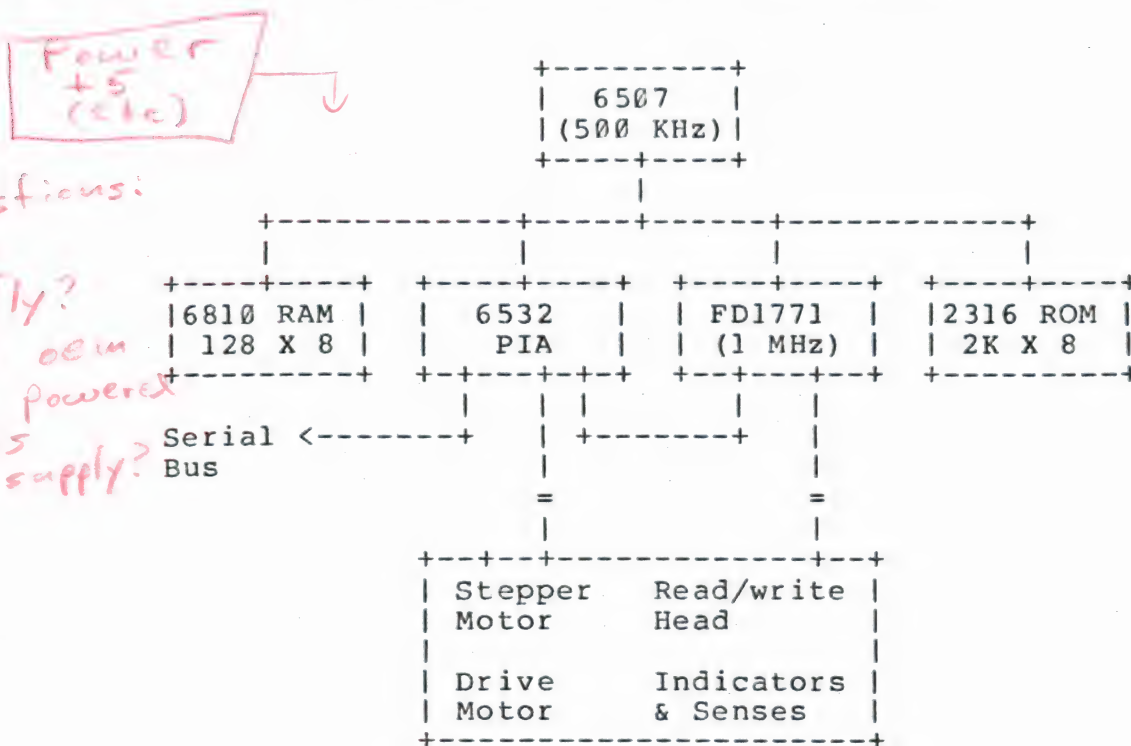
There is a single configuration/option parameter for the Model 810 and that is the logical unit number select. There are a pair of switches at the back of the unit which are used to indicate which of the four possible logical units (1-4) this unit is to become. These switches are to be checked as part of the processing of each command frame.

2. Hardware description

2.1 General description

The Model 810 is comprised of a xx mini-floppy diskette drive and an Atari designed and built controller board mounted together in a single package.

A block diagram of the controller is shown below:



2.2 Functional block diagram

2.3 Controller memory and I/O address assignments

ROM address space = 0800-0FFF (2316).
 RAM address space = 0080-00FF (6801)
 0180-01FF (6532).
 I/O address space = 0000-0003 (FD1771)
 0380-039F (6532).

2.4 I/O interface specification

2.4.1 Parallel I/O ports

All of the I/O for the controller processor is performed via two sets of mapped memory I/O ports; one set is assigned to a FD1771 Floppy Disk Controller chip and the other set is assigned to a 6532 PIA chip. The ports are described further in the sections that follow.

2.4.1.1 FD1771 Floppy Disk Controller chip

The FD1771 chip data access lines expect and provide data in bit complemented form (1 for 0 and 0 for 1); since there is no hardware inversion of the data bus between the 6507 and the FD1771, the inversion must be performed by the controller firmware. The descriptions in this section, as well as those to be found in Appendix B, describe the data as seen inside the FD1771, which is the complement of the data as seen by the 6507.

The port address assignments are shown below:

Address	Direction	Function
0000	READ	Status register.
0000	WRITE	Command register.
0001	RD/WRT	Track register.
0002	RD/WRT	Sector register.
0003	READ	Read data register.
0003	WRITE	Write data register.

Question raised:
Are Address lines inverted? If not, say so explicitly

The description of the FD1771 chip is to be found in Appendix B.

2.4.1.2 6532 Peripheral Interface Adaptor (PIA)

The port address assignments are shown below:

Address	Direction	Function
0380	RD/WRT	Port A.
0381	WRITE	Port A data direction register.
0382	RD/WRT	Port B.
0383	WRITE	Port B data direction register.
0384	(WRITE)	Negative edge detect, disable int. } used?
0385	(WRITE)	Positive edge detect, disable int. } used?
0385	READ	Read and clear interrupt flag. } used?
0386	(WRITE)	Negative edge detect, enable int. } used?
0387	(WRITE)	Positive edge detect, enable int. }
039X	(WRITE)	Interval timer setup.

The bit assignments to the 6532 are shown below, by port:

Port	Bit-#	Direction	Function
0380	7	READ	1 = data request from FD1771.
	6	READ	1 = FD1771 IRQ.
	5	N/A	unused.
	4	READ	1 = diskette write protected.
	3	N/A	unused.
	2	READ	Drive select switch #2 (Note 1).
	1	WRITE	1 = drive motor on/drive select.
	0	READ	Drive select switch #1 (Note 1).
0381	7-0	WRITE	Port 0380 data direction select: 1 = PIA output pin, 0 = PIA input pin.

Note 1 -- The drive select switch values are mapped to logical unit numbers as shown in the table below:

Switch #2 Switch #1 Logical unit

1	1	1
1	0	2
0	0	3
0	1	4

Port Bit-# Direction Function

0382	7	READ	Serial bus data input (inverted).
	6	READ	Serial bus NOT COMMAND- (1=command).
	5	WRITE	Stepper motor phase 4 (Note 2).
	4	WRITE	Stepper motor phase 3 (Note 2).
	3	WRITE	Stepper motor phase 2 (Note 2).
	2	WRITE	Stepper motor phase 1 (Note 2).
	1	READ	Serial bus +5V/READY (0=READY).
	0	WRITE	Serial bus data output (data true). (Note 3).

0383	7-0	WRITE	Port 0382 data direction select: 1 = PIA output pin, 0 = PIA input pin.
------	-----	-------	---

Note 2 -- See section 2.5.1 for a description of the stepper motor characteristics.

Note 3 -- The serial bus data output line is to be switched from a PIA output line to an input line whenever the controller is not outputting data to the serial bus.

WHY? Required, or simply done for no reason?

The ports below all relate to internal functions of the 6532 PIA; the ports designated '(WRITE)' do not operate upon the write data, the command is entirely encoded in the address (as explained in Appendix C).

Port Bit-# Direction Function

0384	(WRITE)	Negative edge detect, disable int.
0385	(WRITE)	Positive edge detect, disable int.
0385	READ	Read and clear interrupt flag.
0386	(WRITE)	Negative edge detect, enable int.
0387	(WRITE)	Positive edge detect, enable int.
039X	(WRITE)	Interval timer setup.

The description of the 6532 chip is to be found in Appendix C.

2.4.2 Serial I/O ports

There are no general purpose serial I/O ports provided in the Model 810.

HUH? why are you saying this?

2.4.3 Interrupts

There is no interrupt capability in the 6507 processor chip.

2.5 Electromechanical components

2.5.1 Head stepper motor

The continuous stepper motor sequence for head inward movement is shown below; the sequence in reverse order provides head outward movement:

Phase4	Phase3	Phase2	Phase1
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1
1	1	0	0
etc.			

Other combinations such as 0 0 0 0, etc. do what?

The minimum delay between steps for reliable operation is 5.25 milliseconds. *Is there a max?*

The reference track is track 0, which may be reached by stepping outward a calculated number of tracks from a known track or by stepping outward some arbitrarily large number that guarantees that track zero is reached. No harm is done by attempting to drive the stepper beyond track 0. *which is actually done*

The inter-track spacing is xx which is equivalent to 1 step.

2.5.2 Diskette drive motor

The operational speed for the drive motor is xx RPM.

The time required for the motor to get to operational speed from a dead stop is 200 milliseconds.

is this how long hardware may take? how long software waits? Include both. Be more specific.

3. Firmware description

3.1 General description

3.2 Command descriptions

3.3 Special considerations

3.4 Data base description

3.5 Module hierarchy

3.6 Procedure descriptions

Appendix A -- Serial I/O Bus Characteristics and Protocol

This appendix describes the electrical characteristics of the Atari 400/800 serial bus, the use of the bus to send bytes of data, the organization of the bytes as "frames" (records), and the overall command sequences which utilize frames and response bytes to provide computer/peripheral communication. All signal direction terminology assumes the viewpoint of the PCS computer (e.g. in/out, receive/send, etc.).

Hardware/electrical characteristics

The Atari 400/800 computer communicates with peripheral devices over a 19,200 baud asynchronous serial port. The serial port consists of a serial DATA OUT (transmission) line, a serial DATA IN (receiver) line and other miscellaneous control lines.

Data is transmitted and received as 8 bits of serial data (LSB sent first) preceded by a logic zero start bit and succeeded by a logic one stop bit. The serial DATA OUT is transmitted as positive logic (+4v = one/true/high, 0v = zero/false/low). The serial DATA OUT line always assumes its new state when the serial CLOCK OUT line goes high; CLOCK OUT then goes low in the center of the DATA OUT bit time.

*But clock isn't useful.
This is confusing.*

An end view of the serial bus connector at the computer or peripheral is shown below (the cable connectors would of course be a mirror image):

2	4	6	8	10	12	
0	0	0	0	0	0	
0	0	0	0	0	0	0
1	3	5	7	9	11	13

where: 1 = computer CLOCK IN.
2 = computer CLOCK OUT.
3 = computer DATA IN.
4 = GND.
5 = computer DATA OUT.
6 = GND.
7 = COMMAND-.
8 = MOTOR CONTROL.
9 = PROCEED-.
10 = +5v/READY.
11 = computer AUDIO IN.
12 = +12v.
13 = INTERRUPT-.

*— which is signal
return & which
is shield?*

*— Define terminology
(-)*

- 1 CLOCK IN is not used by the present O.S. and peripherals. This line can be used in future synchronous communications schemes.

- 2 CLOCK OUT is the serial bus clock. CLOCK OUT goes high at the start of each DATA OUT bit and returns to low in the middle of each bit.
- 3 DATA IN is the serial bus data line to the computer.
- 4 Pin 4 GND is the signal/shield ground line.
- 5 DATA OUT is the serial bus data line from the computer.
- 6 Pin 6 GND is the signal/shield ground line.
- 7 COMMAND- is normally high and goes low when a command frame is being sent from the computer.
- 8 MOTOR CONTROL is the cassette motor control line (high=on, low= off).
- 9 PROCEED- is not used by the present O.S. and peripherals (this line is pulled high). *← passively, inside the 800 ✓*
- 10 +5v/READY indicates that the computer is turned on and ready. This line may also be used as a +5 volt supply of ~~unknown current rating~~ for Atari peripherals only.
- 11 AUDIO IN accepts an audio signal from the cassette.
- 12 Pin 12 is a +12 volt supply of ~~unknown current rating~~ for Atari peripherals only.
- 13 INTERRUPT- is not used by the present O.S. and peripherals (this line is pulled high). *also, ←*

There are no pin reassignments made in the serial bus cable, so pin 3, the computer's DATA IN line, is the peripheral's data output line; and similarly for pin 5.

Serial bus electrical specifications

Peripheral input:

V_{LH} = 2.0v min.
V_{LL} = 0.4v max.

I_{LH} = 20ua. max. @ V_{LH} = 2.0v
I_{LL} = 5ua. max. @ V_{LL} = .4v

Peripheral output (open collector bipolar):

V_{OL} = 0.4v max. @ 1.6 ma.
V_{OH} = 4.5v min. with external 100Kohm pull-up.

V_{cc}/READY input:

V_{LH} = 2.0v min. @ I_{LH} = 1ma. max.
V_{LL} = 0.4v max.
Input goes to logic zero when open.

Bus commands

The bus protocol specifies that all commands must originate from the computer, and that peripherals will present data on the bus only when commanded to. Every bus operation will go to completion before another bus operation is initiated (no overlap). An error detected at any point in the command sequence will abort the entire sequence. *(possibly only at one end — see note at side)*

← Exception: Computer may time out before peripheral & try other I/O before peripheral terminates. This exception is essentially a serial I/O error.

A bus operation consists of the following elements:

- Command frame from the computer.
- Acknowledgement (ACK) from the peripheral.
- Optional data frame to or from the computer.
- Operation complete (COMPLETE) from the peripheral.

COMMAND FRAME

The serial bus protocol provides for three types of commands: 1) data send, 2) data receive and 3) immediate (no data -- command only). There is a common element in all three types, a command frame consisting of five bytes of information sent from the computer while the COMMAND- line is held low. The format of the command frame is shown below:

+-----+
device I.D.
+-----+
command
+-----+
auxilliary #1
+-----+
auxilliary #2
+-----+
checksum
+-----+

The device I.D. specifies which of the serial bus devices is being addressed.

The command byte contains a device dependent command.

The auxilliary bytes contain more device dependent information.

The checksum byte contains the arithmetic sum of the first four bytes (with the carry added back after every addition).

COMMAND FRAME ACKNOWLEDGE

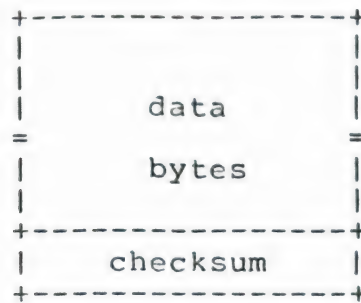
The peripheral being addressed would normally respond to a command frame by sending an ACK byte (\$41) to the computer; if there is a problem with the command frame, the peripheral should not respond.

or respond NAK — depends on specific nature of problem. BE ✓

DATA FRAME

Following the command frame (and ACK) may be an optional data

frame which is formatted as shown below:



This data frame may originate at the computer or at the device controller, depending upon the command. Current device controllers expect fixed length data frames as does the computer, where the data frame length is a fixed function of the device type and command.

The checksum value in the data frame is the arithmetic sum of all of the frame data preceding the checksum, with the carry from each addition being added back (the same algorithm as for the command frame).

In the case of the computer sending a data frame to a peripheral, the peripheral is expected to send an ACK if the data frame is acceptable, and a NAK (\$4E) or nothing if the data frame is unacceptable.

OPERATION COMPLETE

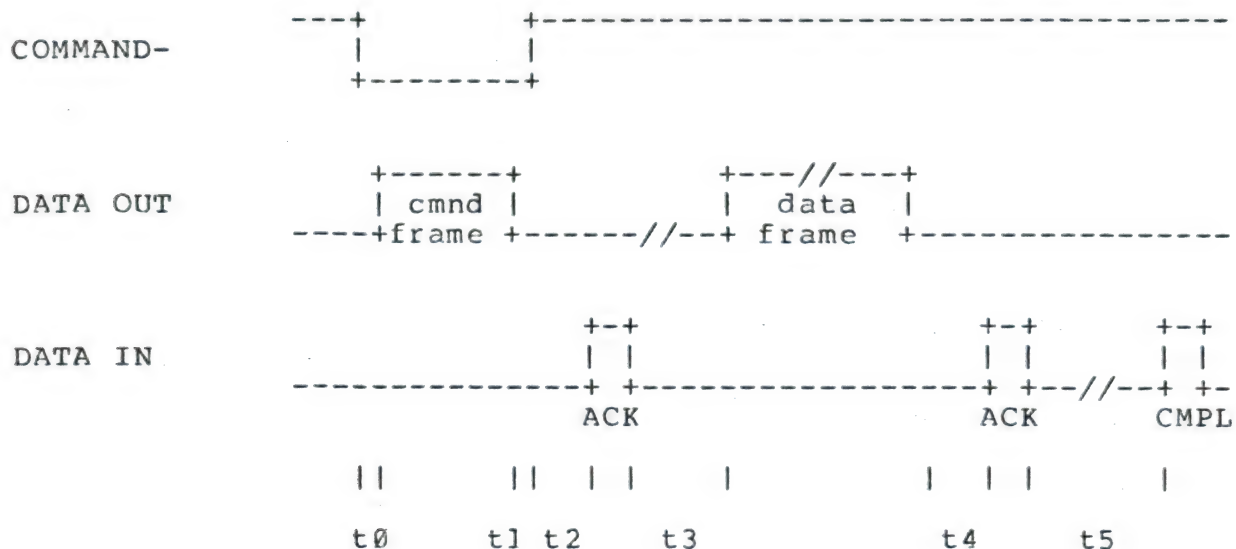
A peripheral is also expected to send an operation COMPLETE byte (\$43) at the time the commanded operation is complete. The location of this byte in the command sequence for each command type is shown in the timing diagrams that follow. If the operation cannot go to normal, error-free completion, the peripheral should respond with an ERROR byte (\$45) instead of COMPLETE.

really???
if so - UGH!

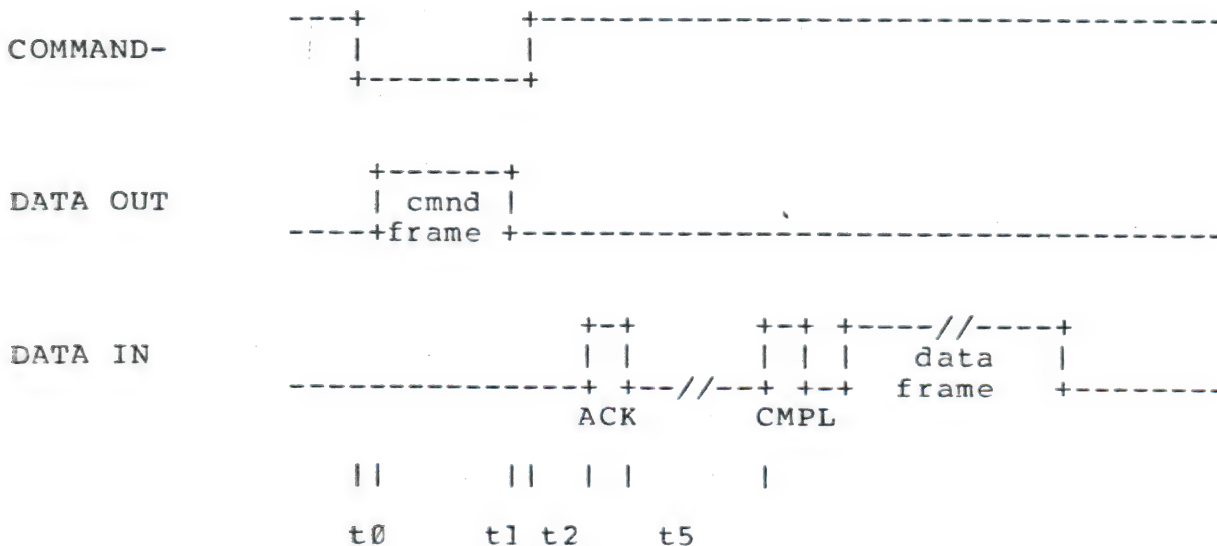
Bus timing

This section provides timing diagrams for the three types of command sequences: data send, data receive and immediate.

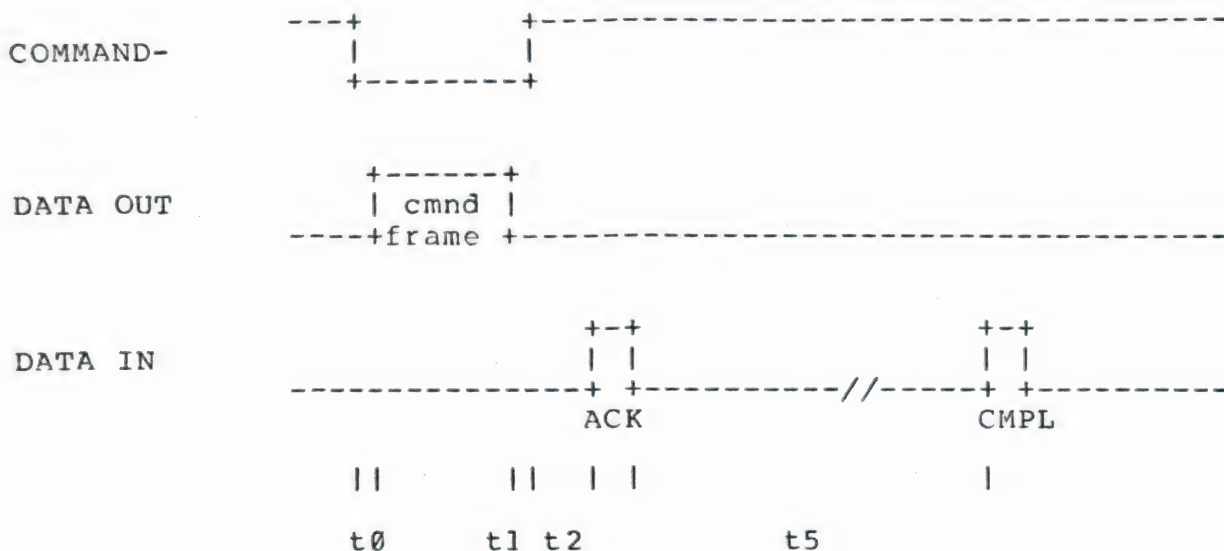
COMPUTER DATA SEND (PERIPHERAL DATA RECEIVE) sequence:



COMPUTER DATA RECEIVE (PERIPHERAL DATA SEND) sequence:



IMMEDIATE sequence:



t0 is the delay between the lowering of COMMAND- and the transmission of the first byte of the command frame. The computer generates this delay.

computer t0 (min) = 750 usec.
computer t0 (max) = 1600 usec.

peripheral t0 (min) = ??
peripheral t0 (max) = ??

t1 is the delay between the transmission of the last bit of the command frame and the raising of the COMMAND- line. This delay is generated by the computer.

computer t1 (min) = 650 usec.
computer t1 (max) = 950 usec.

peripheral t1 (min) = ??
peripheral t1 (max) = ??

t2 is the delay between the raising of COMMAND- and the transmission of the ACK byte by the peripheral. The peripheral generates this delay.

computer t2 (min) = 0 usec.
computer t2 (max) = 16 msec.

peripheral t2 (min) = ??
peripheral t2 (max) = ??

t3 is the delay between the receipt of the last bit of the ACK byte and the transmission of the first bit of the data frame by the computer. The computer generates this delay.

computer t3 (min) = 1000 usec.
computer t3 (max) = 1800 usec.

peripheral t3 (min) = ??
peripheral t3 (max) = ??

t4 is the delay between the transmission of the last bit of the data frame and the receipt of the first bit of the ACK byte by the computer. The peripheral generates this delay.

computer t4 (min) = 850 usec.
computer t4 (max) = 16 msec.

peripheral t4 (min) = ??
peripheral t4 (max) = ??

t5 is the delay between the receipt of the last bit of the ACK byte and the first bit of the COMPLETE byte by the computer. The peripheral generates this delay.

computer t5 (min) = 250 usec.
computer t5 (max) = 255 sec. (handler dependent)

peripheral t5 (min) = ??
peripheral t5 (max) = N/A

Serial bus special character values

ACK = \$41.
NAK = \$4E.
COMPLETE = \$43.
ERR = \$45.

*Note: newer peripherals
send other values, but
this does not apply to
disk.*

Serial bus device I.D. to logical unit mapping (Model 810)

Unit 1 = \$31.
Unit 2 = \$32.
Unit 3 = \$33.
Unit 4 = \$34.

Appendix B -- FD1771 Floppy Disk Controller Chip Description

Appendix C -- 6532 PIA Chip Description

Appendix D -- 6507 Microcomputer Description

Appendix E -- Model 810 Controller Schematics